อาจหาญ สัตยารักษ์

วันนี้เรามาพูดถึงเรื่อง CAI กันสักนิด (CAI คืออะไร คิดว่าผู้อ่านคงทราบแล้วจากบทความอื่น จึงของดอธิบาย) โดยขอกล่าวถึงเกร็ดเล็ก ๆ น้อยๆในการสร้างโปรแกรมคอมพิวเตอร์ประเภท CAI ว่าในฐานะคนเขียนโปรแกรมมองแนวคิดเกี่ยวกับเรื่องนี้อย่างไรบ้าง แต่เนื่องจากผู้เขียนเป็นครูสอนในระดับโรงเรียนมัธยมศึกษา จึงทำให้รู้จักกับคนเขียนโปรแกรมในประเภทเดียวกันเกือบทั้งหมด ดังนั้น แนวคิดที่จะได้อ่านต่อไปนี้จึงเป็นแนวคิดตามที่ผู้เขียนสัมผัสกับกลุ่มคนเขียนระดับดังกล่าวเป็นส่วนใหญ่ อาจไม่ตรงกับทฤษฎีที่ผู้รู้ทั้งหลายร่ำเรียนมาเลยก็ได้ **เพราะส่วนมากพวกเรามองผลการทำงานของโปรแกรมมากกว่ามองว่าทำโปรแกรมในภาษาใด** ดังนั้นมีสิ่งใดที่เป็นแนวคิดที่ไม่ตรงกับทฤษฎีใดๆ ก็โปรดเข้าใจว่านั้นคือแนวคิดใหม่ของพวกเรานั่นเอง

มาติดตามกันต่อไป...

แนวการสร้าง CAI

1. นักเขียนโปรแกรม (Programmer)

การสร้าง CAI ในระดับมัธยมศึกษามักเกิดจากคน 2 กลุ่ม คือ **ครูสอนวิทยาศาสตร์ (โดยเฉพาะครูสอนฟิสิกส์) กับ ครูสอนคณิตศาสตร์** เพราะอย่างน้อยครูพวกนี้ก็มีความคิด ความสนใจด้านวิทยาศาสตร์ และการคำนวณ เป็นทุนอยู่ก่อนแล้ว ความกลัวเครื่องคอมพิวเตอร์จึงมีน้อย (เป็นเรื่องจริงนะ ที่มีครูอาจารย์บางส่วนยังกลัวเครื่องคอมพิวเตอร์อยู่ ซึ่งก็คือเหตุหนึ่งที่การใช้เครื่องยังอยู่ในวงแคบๆ)

2. เรื่องที่เขียน

แน่นอน...คนที่เขียนโปรแกรมแล้วไม่สามารถแลกเปลี่ยนโปรแกรมนั้นให้ออกมาเป็นรายได้ตอบแทน สิ่งแรกที่คนเขียนคำนึงก่อนสิ่งอื่นใดก็คือ **ทำให้ตนเองเอาไปใช้สอน** เรื่องคนอื่นเอาไปใช้เป็นเรื่องทีหลัง เรื่องที่พวกเราเลือกมาเขียนจึงต้องเอาที่ใกล้ตัวก่อน ผลก็คือโปรแกรมที่เขียนส่วนใหญ่ก็เป็นว่า ถ้าครูฟิสิกส์เขียนก็นึกถึงวิชาฟิสิกส์ (อย่างผู้เขียนก็เป็นตัวอย่างหนึ่งที่ผู้อ่านคงพอรู้อยู่บ้างแล้ว) ถ้าครูคณิตศาสตร์เขียนก็หนีไม่พ้นวิชาคณิตศาสตร์ (วิชาคอมพิวเตอร์ซึ่งมี 2 รายวิชา คือ ค 031 และ ค 032 ก็อยู่ในวิชาคณิตศาสตร์)

ดังนั้นโปรแกรม CAI ที่ระบาดในกลุ่มครูมัธยมศึกษาจึงเป็นโปรแกรม 2 วิชานี้เป็นส่วนใหญ่ (ความจริงของต่างประเทศก็เกือบไม่ต่างกันนัก) ส่วนมากก็ก๊อบปี้กันฟรีๆแบบแลกเปลี่ยนกัน

สถาบันส่งเสริมการสอนวิทยาศาสตร์และเทคโนโลยี (สสวท) เคยจัดการแสดงผลงานประเภทนี้มาแล้ว 2 ครั้ง คือปี 2529 และปี 2530 โปรแกรมส่วนมากก็เป็น 2 วิชานี้ทั้งหมด ครูอาจารย์อื่นๆที่ไม่ได้เข้าร่วมก็ติดต่อได้โดยตรงที่ สสวท ดังกล่าว หรือท่านมีโปรแกรมเด็ดๆอยากจะไปแสดงกับคนอื่นเขาบ้างก็เตรียมตัวได้เลย เพราะประมาณต้นเดือนพฤษภาคม 2531 เขาจะจัดให้มีการแสดงผลงาน CAI ของวิชาในระดับไม่สูงกว่ามัธยมศึกษาตอนปลาย ครั้งที่ 3 (รายละเอียดโปรดติดต่อกับ สสวท เองนะครับ)

3. ภาษาที่ใช้เขียน

เนื่องจากพวกเรามักเป็นประเภทไม่เคยเรียนคอมพิวเตอร์ในสถานศึกษากันมาก่อน ต้องศึกษาด้วยตนเอง และเผอิญในช่วงแรกๆนั้น หนังสือในท้องตลาดมักเป็นภาษาเบสิกเสียส่วนมาก ดังนั้นภาษาที่ครูมัธยมศึกษาใช้ในการเขียนโปรแกรมจึงหนีไม่พ้นภาษาเบสิก โดยเฉพาะแอปเปิ้ลซอฟต์เบสิกกันเป็นส่วนมาก (ถ้าผู้รู้ท่านใดบอกว่าใช้ภาษาเบสิกไม่ดี ไม่เหมาะสม ก็คงต้องโทษหนังสือพวกนั้นก็แล้วกัน ?)

จะมีการใช้ภาษาอื่นมาเกี่ยวข้องบ้างก็แค่การใช้เสริมเท่านั้น

ดังนั้นงานการเขียนโปรแกรมจึงอาจจะแสดงผลบนจอภาพได้อย่างไม่ต่อเนื่องนัก แต่พวกเราก็คิดว่า ยังดีกว่าการเขียนรูปที่นิ่งๆบนกระดานดำให้เด็กดูแน่ๆ อย่างน้อยก็ทำให้รู้สึกว่าตนเองเป็นครูที่ทันสมัยกับเขาขึ้นมาหน่อย

แต่ปัจจุบัน พวกเราก็หันมาเขียนกันบนภาษาอื่นๆมากขึ้น แสดงให้เห็นว่าเราเริ่มพัฒนาไปตามกาลเวลาเอง

4. การวิจัยผลงาน

โดยส่วนมากพวกเราสร้างโปรแกรมขึ้นมาเพื่อช่วยการสอน และพวกเราก็ไม่ค่อยมีความรู้ในการวิจัยกันมากนัก โปรแกรมส่วนใหญ่จึงยังไม่ได้ทำการวิจัยผลงานว่าดี เหมาะสม มีผลสัมฤทธิ์แค่ไหน ควรแก้ไขโปรแกรมหรือไม่อย่างไร

ส่วนมากก็ประเมินเอาเองว่าดีจากการที่พบว่าเด็กสนใจการเรียนมากขึ้น (แต่ไม่รู้ว่าสนใจโปรแกรม หรือสนใจเครื่องคอมพิวเตอร์เท่านั้น)

กราฟิกบนงาน CAI

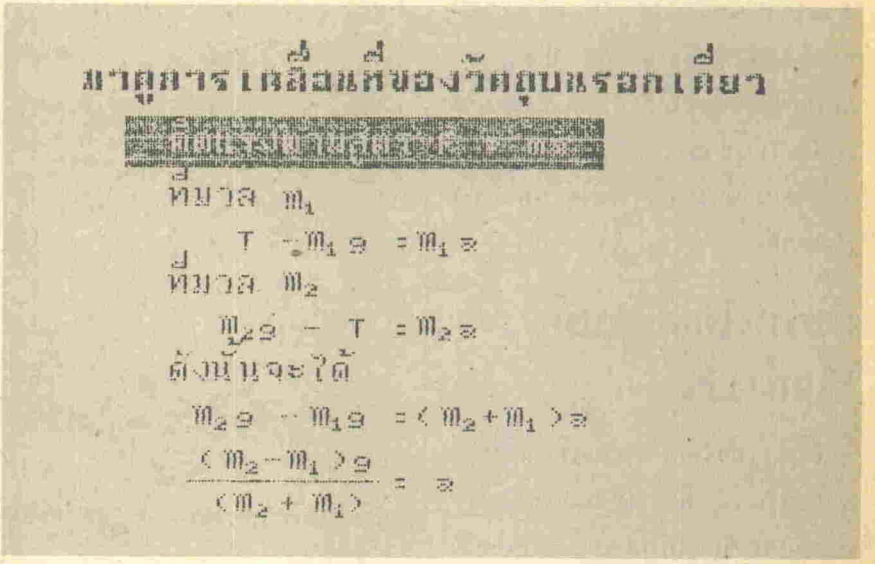
มีความจำเป็นแค่ไหน ในการใช้กราฟิกในการเขียนโปรแกรมช่วยการเรียนการสอน ไม่มีกราฟิกเลยมันไม่เป็น CAI หรือไง หรือถ้ามีกราฟิกแน่ๆ กราฟิกนั้นควรเขียนอย่างไรจึงจะเป็นกราฟิกที่เข้ากันกับโปรแกรม นี่คือปัญหาที่น่าสนใจไม่น้อย

ลองมาดูภาพที่ตัดมาจากโปรแกรมช่วยการเรียนการสอนเรื่องหนึ่ง โดยในโปรแกรมนี้ไม่มีภาพกราฟิกประกอบ คือรูปที่ 1 กันสักนิดหนึ่ง

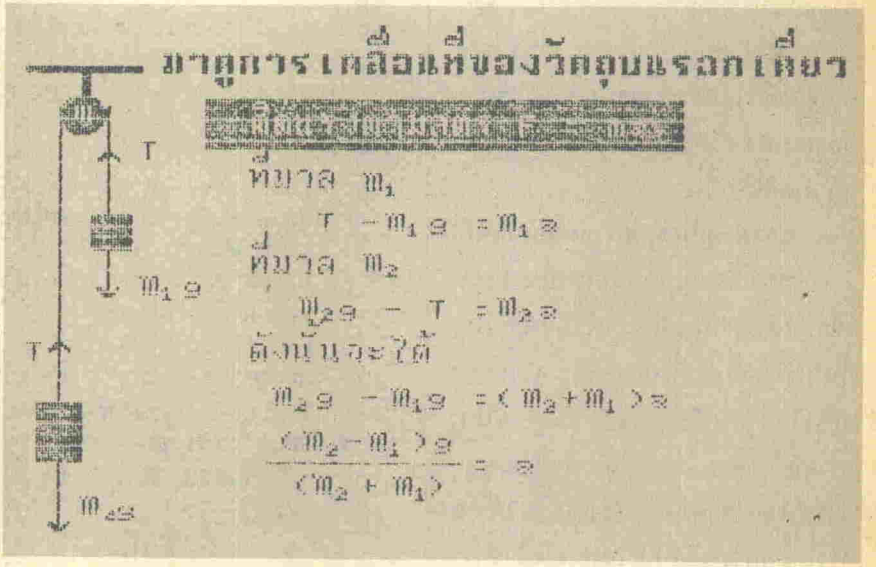
จากรูปดังกล่าวผู้อ่านคงต้องงงแน่ๆ เพราะตรงไหนคือมวล m1 ตรงไหนคือมวล m2 แรง T มาจากไหน ดังนั้นถ้าโปรแกรมช่วยการเรียนการสอนเป็นแบบนี้ งานด้านกราฟิกเป็นสิ่งที่จำเป็นต่อการนำมาช่วยให้โจทย์ แนวคิด คำอธิบาย มีความกระจ่างขึ้นกว่าที่มีแต่ตัวหนังสือ (text) เพียงอย่างเดียว

ลองมาดูรูปที่ 2 ซึ่งมีภาพประกอบอย่างสมบูรณ์ ก็คงจะเห็นจริง

ดังนั้นเราจึงสรุปกันง่ายๆว่า **กราฟิกนั้นจำเป็นมากสำหรับ CAI ในวิชาวิทยาศาสตร์และคณิตศาสตร์** เพราะโดยส่วนมากเราก็เขียนโปรแกรมช่วยอธิบายสิ่งที่เรามองไม่เห็นจากการทดลอง (เช่น การวิ่งของรังสีต่างๆ) หรือในบทเรียน



รูปที่ 1 โปรแกรมที่มีแต่เท็กซ์



รูปที่ 2 โปรแกรมที่มีทั้งเท็กซ์และกราฟิก

เป็นภาพนิ่งเราก็อยากทำให้เกิดการเคลื่อนไหวหรือมีการทดลองแต่เกิดไวเกินไปจนมองไม่ทัน (เช่นแสงเคลื่อนที่) เป็นต้น ภาพลักษณะอย่างนี้แหละที่พวกเราเลือกมาทำ CAI ก็เลยต้องใช้กราฟิกอย่างหลีกเลี่ยงไม่ได้

ความจริงภาพที่เกิดขึ้นจริงๆจากรูปที่ 1 และ 2 นั้น มีการเคลื่อนที่ของวัตถุทั้งสองให้ดูก่อนที่จะมีการสรุปในภาพสุดท้าย

แสดงว่าเขียนกราฟิกแล้วยังไม่พอใจเช่นนั้นหรือ

คงเป็นเช่นนั้น...เพราะบางทีการเคลื่อนไหวก็ช่วยให้มองเห็นภาพที่ดีขึ้น กระจ่างแจ้งต่อความเข้าใจมากขึ้น จึงเป็นผลให้พวกเราต้องมาสนใจเกี่ยวกับเรื่องการพยายามทำให้ภาพเกิดการเคลื่อนไหวขึ้นมาอีก เรื่องยุ่งยากจึงยิ่งเพิ่มทวีคูณ

ทำภาพให้เคลื่อนที่ได้อย่างไร

การทำให้เกิดภาพเคลื่อนที่หรือเคลื่อนไหวนั้นที่จริงเป็นเรื่องยากมากสำหรับคนเขียนโปรแกรมมืออ่อนอย่างพวกเรา เพราะมันลำบากจริงๆที่จะให้โปรแกรมทำงานอย่างที่เราต้องการที่ยุ่งยากมากๆก็เพราะตำราคอมพิวเตอร์แต่ละเล่มมักไม่เขียนเกี่ยวกับเรื่องนี้เสียด้วย เราก็ได้แต่พยายามมองหาวิธีการของเราเอง ใครคิดได้ก็เล่าก็ต่อๆไป แนวคิดก็เพิ่มขึ้น

ลองมาดูกันสักหนึ่งแนวคิดก่อน

สมมติเรามีวัตถุก้อนหนึ่งวางบนโต๊ะ แล้วเราก็อยากจะให้วัตถุดังกล่าวนั้นมีการเคลื่อนที่ไปตามโต๊ะ เราจะทำอย่างไร เขียนเป็น Shape Table ดีไหม บางคนก็ว่าดีแน่ เพราะใช้คำสั่ง DRAW, XDRAW วาดภาพหรือลบภาพให้เกิดการเคลื่อนไหวได้ แต่...บางคนก็แย้งว่าดีนะดีหรอกถ้าวัตถุนั้นขนาดไม่ใหญ่นัก เพราะถ้าวัตถุมีขนาดใหญ่แล้วการวาดภาพ ลบภาพ ก็ไม่ค่อยน่าดูเหมือนกัน มาดูแนวคิดแบบนี้ดีกว่า

ลองดูรูปที่ 3 กันก่อนดีไหม

สมมติวัตถุมีขนาดดังรูป คือค่าบนแกนราบ (ค่า X) เริ่มที่ 10 จบที่ 30 ส่วนค่าบนแกนตั้ง (ค่า Y) บนสุดอยู่ที่ 30 ล่างสุดอยู่ที่ 50 และเราอยากจะให้วัตถุเคลื่อนที่ไปทางขวา (ค่า X เพิ่มขึ้น)

เราก็ทำดังนี้

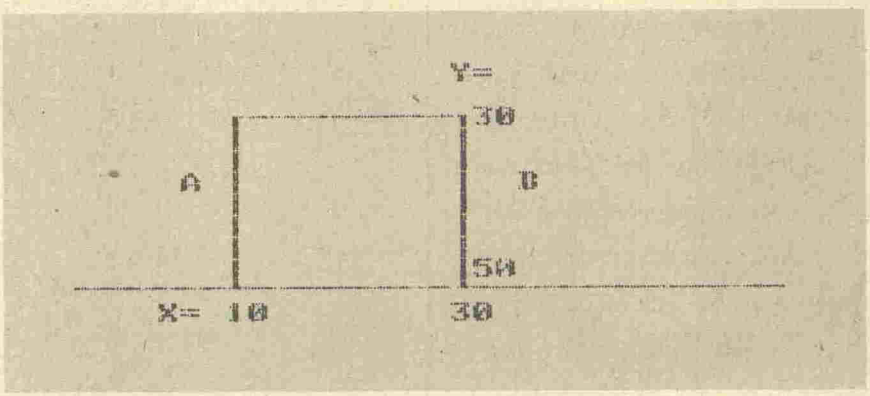
1. กำหนดค่า K = 0

1. ลบภาพทางด้านซ้าย (คือลบเส้นตรง 1 เส้นที่ค่า X = 10 + K = 10) ให้ปรากฏว่าวัตถุขยับไปทางขวาหนึ่งแนวโดยลบจากบนสุด (ค่า Y = 30) ไปจนถึงล่างสุด (ค่า Y = 50)

2. วาดภาพทางด้านขวา (คือวาดเส้นตรง 1 เส้นที่ค่า X = 31 + K = 31) ให้ปรากฏภาพวัตถุขยับไปทางขวาหนึ่งแนวเช่นกัน

3. ขยับจุดลบภาพ, วาดภาพไปทางขวาหนึ่งแนว (K = K + 1) แล้วทำซ้ำกับข้อ 1 และข้อ 2 ไปจนกว่าจะพอใจ

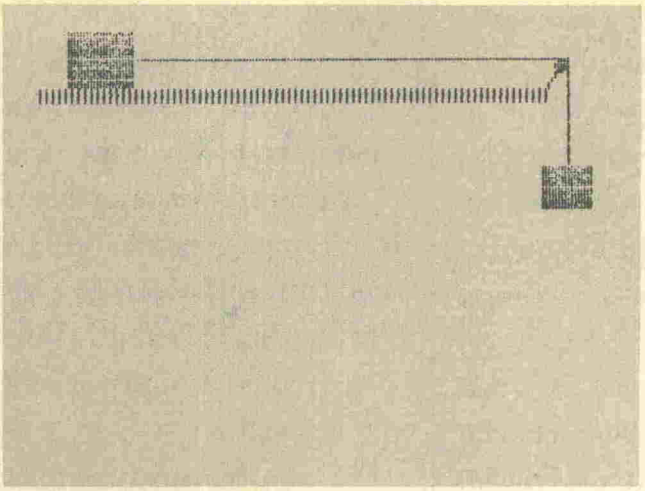
แค่นี้เราก็ได้ภาพวัตถุเคลื่อนที่ไปทางขวาแล้ว และผลปรากฏว่าการเคลื่อนที่นุ่มนวลกว่าวิธีใช้ Shape Table เสียด้วย โปรแกรมการทำงานจริงดังรูปที่ 4 ซึ่งในที่นี้เป็นการเคลื่อนที่ของวัตถุ



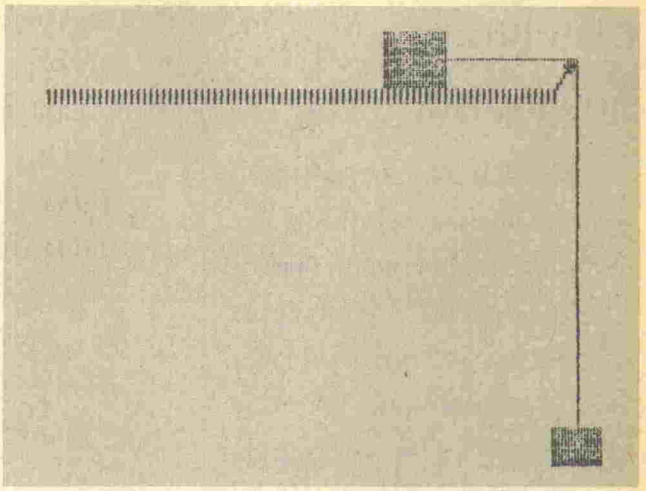
รูปที่ 3 ภาพแสดงแนวคิด เมื่อต้องการให้วัตถุเคลื่อนที่

```
10  REM  By Arthan Sattayrak
20  REM    APPLESOFT BASIC
30  HGR2 : HCOLOR= 5
40  FOR K = 0 TO 5: HPLOT 0,51 + K TO 160,51 + K: NEXT
50  HCOLOR= 3: HPLOT 160,51 TO 164,42
60  FOR K = 0 TO 2: HPLOT 164 - K,40 + K TO 164 + K,40 + K
70  HPLOT 164 - K,45 - K TO 164 + K,45 - K: NEXT
80  FOR K = 10 TO 30: HPLOT K,30 TO K,50: NEXT
90  FOR K = 0 TO 15: HPLOT 158 + K,80 TO 158 + K,95: NEXT
100  HPLOT 30,40 TO 166,40 TO 166,80
110  FOR K = 0 TO 95
120  HCOLOR= 0: HPLOT 10 + K,30 TO 10 + K,50
130  HPLOT 158,80 + K TO 173,80 + K
140  HCOLOR= 3: HPLOT 31 + K,30 TO 31 + K,50
150  HPLOT 158,96 + K TO 173,96 + K
160  HPLOT 166,75 + K TO 166,80 + K
170  NEXT
```

รูปที่ 4 โปรแกรมแสดงการวาดภาพวัตถุ 2 ชิ้นบนรอกเดียวตายตัว



รูปที่ 5 รูปวัตถุตอนเริ่มต้น



รูปที่ 6 เมื่อวัตถุเคลื่อนที่แล้ว

2 ชิ้นที่ผูกติดกันด้วยเชือกแล้วคล้องกับรอกเดี่ยวตายตัว ซึ่งจะมีผลทำให้วัตถุทั้งสองมีการเคลื่อนที่เกิดขึ้น

ซึ่งช่วงบรรทัดที่ 10-100 จะให้ภาพปรากฏบนจอเช่นเดียวกับรูปที่ 5 ส่วนช่วงบรรทัดที่ 110-170 ก็คือส่วนที่ทำให้เกิดภาพวัตถุเคลื่อนที่และจบลงด้วยการเกิดภาพ ดังรูปที่ 6

นี่เป็นแนวคิดหนึ่งที่ง่ายๆ แต่ได้ผลเป็นที่น่าพอใจมาก ลอง RUN โปรแกรมดูท่านจะเห็นจริงด้วย 1000% เลยทีเดียวนะ

ส่วนท่านที่มีเครื่อง 16 บิตก็อาจใช้โปรแกรม HBASIC, BASICA, GWBASIC ได้ เพียงดัดแปลงโปรแกรมนี้เสียนิดหน่อยโดยแก้ไข SYNTAX เสียใหม่ให้ตรงกับของภาษานั้นๆก็จะทำงานได้ผลเช่นกัน

คำที่ใช้นั้นจะเทียบได้ดังข้างล่างนี้

```
B>
 { PASCAL   By Arthan Sattayarak }

 VAR  k : Integer;
  BEGIN
     Graphmode;
     for k := 0 to 5 do
      begin;
        draw(0,51+ k,160,51+k,1);
      end;
       draw(160,51,164,42,1);
         for k := 0 to 2 do
           begin;
            draw(164-k,41+k,164+k,41+k,1);
            draw(164-k,45-k,164+k,45-k,1);
           end;
         for k :=10 to 30 do
           begin;
             draw(k,30,k,50,2);
           end;
               for k := 0 to 15 do
                 begin;
                  draw(158+k,30,158+k,95,3);
                 end;
               draw(30,40,166,40,1);
               draw(166,40,166,80,1);
          for k := 0 to 95 do
            begin;
              draw(10+k,30,10+k,50,0);
              draw(158,80+k,173,80+k,0);
              draw(31+k,30,31+k,50,2);
              draw(158,96+k,173,96+k,1);
              draw(166,75+k,166,80+k,3);
            end
   END.
```

รูปที่ 7 โปรแกรมภาษาปาสคาล ทำงานแบบเดียวกับโปรแกรมเบสิกในรูปที่ 4

แอปเปิ้ลซอฟต์เบสิก	ไอบีเอ็ม
HGR2	SCREEN 1
HPLOT 0,51+K TO 160,51+K	LINE(0,51+K)-(160,51+K),2
HCOLOR=0:HPLOT 10+K,30 TO 10+K,50	LINE(10+K,30)-(10+K,50),0
HCOLOR=3:HPLOT 31+K,30 TO 31+K,50	LINE(31+K,30)-(31+K,50),3

หรือจะลองเขียนด้วยภาษาปาสคาล (Pascal) ก็จะได้ดังรูปที่ 7 ซึ่งในที่นี้ผู้เขียนพยายามเขียนให้ตรงกับรูปที่ 4 อยู่แล้ว จึงคิดว่าท่านก็สามารถศึกษาเปรียบเทียบได้ไม่ยากนัก

การเคลื่อนที่ที่ตรงความเป็นจริง

แต่ถ้าท่านสังเกตการทำงานของโปรแกรมในรูปที่ 4 หรือรูปที่ 7 ก็จะเห็นว่ามันไม่ตรงกับความเป็นจริง เพราะในโปรแกรมดังกล่าวนี้วัตถุทั้งสองจะเคลื่อนที่ด้วยความเร็วคงที่ ซึ่งที่จริงแล้ววัตถุทั้งสองต้องเคลื่อนที่ด้วยความเร่งคงที่ (a คงที่ หรือความเร็วเพิ่มขึ้นเรื่อยๆ) ตามแนวคิดของ "กฎการเคลื่อนที่ของนิวตัน"

ความไม่พอใจการทำงานของโปรแกรมก็เกิดขึ้นอีก

สูตรที่เราใช้ในบทเรียนจึงต้องเข้ามามีบทบาทต่อการเขียนโปรแกรมมากขึ้น กล่าวคือ Listing ของโปรแกรมสามารถนำไปอธิบายให้เด็กที่พอรู้ภาษาของคอมพิวเตอร์เข้าใจในการเกิดภาพแต่ละภาพว่าเป็นไปตามทฤษฎีที่เรียนมาหรือไม่อย่างไร

ลองยกตัวอย่าง **ของตกอย่างอิสระในแนวดิ่ง** นี่ดูนะ

ในการเรียนเรื่องนี้บอกว่าเมื่อวัตถุถูกปล่อยให้ตกอย่างอิสระ (ความเร็วต้น u มีค่าเท่ากับศูนย์) นั้น ระยะทาง s ที่วัตถุเคลื่อนที่นับจากจุดปล่อยจะเปลี่ยนไปตามเวลา t ด้วยความเร่ง g ที่มีค่า 9.8 เมตร/วินาที/วินาที โดยมีสูตรดังนี้

$$s = ut + \frac{1}{2}gt^2$$

$$s = \frac{1}{2}gt^2 \quad \text{เมื่อค่า } u = 0$$

ดังนั้นเมื่อเขียนเป็นสูตรในโปรแกรมคอมพิวเตอร์ก็จะได้ดังนี้

```
S = 1/2*G*T*T
```

แต่ถ้าเรากำหนดค่า g ให้ตรงความเป็นจริงคือ 9.8 วัตถุก็จะตกเร็วเกินไป เราจึงให้ค่า g ในการทำงานของโปรแกรมผิดความจริงไป คือค่าลดลง 10 เท่า คือใช้ 0.98 หรือ 1 แทน (เปรียบเสมือนกับว่าแรงดึงดูดบนพื้นผิวคอมพิวเตอร์มีค่าเป็น 0.1 ของพื้นโลกก็ได้)

ท่านลองดูโปรแกรมของตกในรูปที่ 8 จะพบว่าเราได้ใส่สูตรในบทเรียนลงไปให้เครื่องคำนวณเพื่อใช้ในการกำหนดตำแหน่งวาดรูป, ลบรูปของตกเอง ผู้อ่านจะคีย์บนเครื่องใดก็ตามใจ ช่วงแรกเป็นของแอปเปิ้ลซอฟต์เบสิก ถัดมาเป็นของ 16 บิต (บนเทอร์โบเบสิก) ซึ่งถ้าท่านจะเขียนแบบเบสิกธรรมดาก็ให้ใส่หมายเลขบรรทัดลงไปตามลำดับก็จะทำงานได้เช่นกัน

ไม่มีรูปผลของการทำงานให้ดูนะ เพราะอยากให้คีย์ดูเอง โปรแกรมก็สั้นๆ คีย์สบายๆ

มาลองดูกันสักอีกแนวคิดหนึ่ง

คราวนี้ลองมาดู **"การตกของหยดน้ำมันที่มีประจุไฟฟ้าในแนวดิ่ง ผ่านบริเวณที่มีสนามไฟฟ้าทิศทางลงในแนวดิ่ง"** ซึ่งตามทฤษฎีบอกว่า จะมีแรงกระทำต่อประจุไฟฟ้านั้นดังรูปที่ 9 โดยรูปทางซ้ายคือแรงที่เกิดกับประจุไฟฟ้าบวก (+) รูปทางขวาคือแรงที่เกิดกับประจุไฟฟ้าชนิดลบ (−)

แรง mg คือแรงเนื่องจากน้ำหนักของหยดน้ำมันมีทิศลงในแนวดิ่ง

แรง F (ซึ่งเท่ากับ qE) คือแรงเนื่องจากประจุไฟฟ้าอยู่ในสนามไฟฟ้า

ดังนั้นหยดน้ำมันจึงจะต้องเคลื่อนที่ด้วยแรง mg+qE หรือเคลื่อนด้วย

```
10  REM  By Arthan Sattayarak
20  REM    APPLESOFT BASIC
30  FOR K = 24576 TO 24592
40  READ A: POKE K,A: NEXT
50  HGR2 : POKE 232,0: POKE 233,96: ROT= 0: SCALE= 1
60  HPLOT 30,149 TO 260,149
70 G = 1
80  FOR T = 0 TO 17 STEP .5
90 S = 0.5 * G * T * T
100  DRAW 1 AT 160,S: XDRAW 1 AT 160,S
110  NEXT
120  DRAW 1 AT 160,S
130  DATA 1,0,4,0,41,117,63,63,46
140  DATA 45,53,63,63,14,45,5,0
```

```
' Turbo Basic By Arthan Sattayarak
screen 1 : color 0,0
line(30,149)-(260,149),3
g = 1 : ball$ ="r2f1l4d1r4d1l4f1r2"
for k = 0 to 17 step .5
s = 0.5*g*t*t
draw"bm160,"+str$(int(s))+"c1"+ball$
draw"bm160,"+str$(int(s))+"c0"+ball$
next
draw"bm160,"+str$(int(s))+"c1"+ball$
```

รูปที่ 8 โปรแกรมการแสดงลูกบอลตกในแนวดิ่ง แอปเปิ้ลซอฟต์เบสิก และเทอร์โบเบสิก

ความเร่งขนาดเท่ากับ (mg+qE)/m ตามสูตรการเคลื่อนที่ของนิวตัน F = ma นั่นเอง

เมื่อดูระยะทาง s ที่จะเคลื่อนที่นับเวลา t ตั้งแต่หลุดเข้ามาในสนามไฟฟ้า จะได้สูตร (เขียนแบบภาษาคอมพิวเตอร์) ดังนี้

```
s = u*t+1/2*a*t*t
s = u*t+1/2*(m*g+q*E)/
    m*t*t
```

ถ้าเราเขียนโปรแกรมแบบให้เครื่องคิดการแสดงผลเอาเอง เราก็คงต้องให้เครื่องคอมพิวเตอร์คิดสถานการณ์ขึ้นเอง กล่าวคือให้ตัวแปร m, q, E เป็นตัวแปรชนิดสุ่ม (Random) ขึ้นมาจากการทำงาน (ค่า q นั้นเครื่องต้องสุ่มมา 2 กรณีเสมอ คือ 1. สุ่มว่าเป็นประจุชนิดบวกหรือลบ 2. สุ่มว่ามีประจุไฟฟ้ากี่ประจุ) ผลที่ดีก็คือแม้แต่คนเขียนโปรแกรมเองก็ไม่มีทางรู้ว่าการแสดงผลจะเป็นอย่างไร ซึ่งกรณีอย่างนี้ก็ทำให้โปรแกรมของเราดูดีขึ้น

โปรแกรมในรูปที่ 10 คือตัวอย่างของการทำงาน (เขียนแบบเทอร์โบเบสิก) ซึ่งคงเข้าใจได้ไม่ยาก หรืออาจจะดัดแปลงมาเป็นเบสิกธรรมดาของ 16 บิตก็ได้ แต่...คงเช่นเดิมเพื่อให้เหมาะกับการทำงานเราจึงไม่ได้ใช้ค่าประจุไฟฟ้าตามความจริงด้วยเช่นกัน

โปรดพิจารณาจากการทำงานของโปรแกรมเอาเองนะ

แล้วท่านจะไม่ลองแก้ไขโปรแกรมในรูปที่ 4 หรือรูปที่ 7 ให้แสดงภาพการเคลื่อนที่ที่ตรงกับความเป็นจริงสักหน่อยหรือ

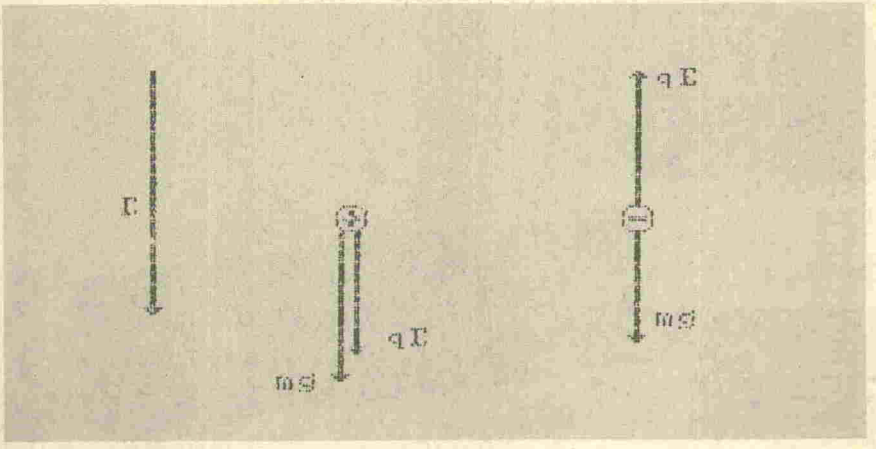
จะเห็นว่าพวกเราครูมัธยมศึกษาที่พยายามผลิตโปรแกรม CAI ขึ้นมาใช้กันเองนี้ ก็ใช้ความรู้เท่าที่พอจะมีอยู่มาพัฒนาโปรแกรมของเราตลอดเวลา พยายามหาทางทำให้โปรแกรมทำงานอย่างถูกต้องมากที่สุด แม้เราต้องไฝ่หาเทคนิคเหล่านี้เอาเองก็ตาม จนบางคนมีความรู้และเทคนิคในการเขียนที่ช่ำชองไม่น้อย ไม่ว่าจะเป็น อาจารย์ที่เชียงใหม่ ขอนแก่น ชัยนาท กรุงเทพฯ ฯลฯ

เมื่อนึกถึงเวลาที่ผ่านไปนานหลายปีที่เล่นคอมพิวเตอร์กันมา จึงคิดว่ายังมีผลงานด้านโปรแกรม CAI ดีๆอีกมากที่ยังหลบซ่อนตัวอยู่ในโรงเรียนของตนเอง ยังไม่ได้เผยโฉมเด่นๆนั้นมาอวดชาวบ้านร้านตลาดเลย **ผู้เขียนจึงอยากจะชวนพวกเราครูมัธยมศึกษาหรือผู้อ่านที่สนใจเรื่อง CAI เสนอผลงานของท่านผ่านมาทางวารสารต่างๆให้มากขึ้น**

ยังไม่รู้เลยว่าเขียนมาตั้งนานนี้เนื้อหาตรงกับ **"กราฟิกกับงาน CAI ในระดับมัธยมศึกษา"** หรือเปล่า แต่...ก็คิดว่าผู้อ่านคงได้อะไรไม่มากก็น้อยจากบทความต่างๆ เนื่องในฉบับ CAI ของวารสารไมโครคอมพิวเตอร์นี้

คงต้องบอกว่าพบกันใหม่ในฉบับต่อไป ด้วยโปรแกรมช่วยการเรียนการสอนวิชาฟิสิกส์ ที่ผู้เขียนถนัด

อย่าลืม...ส่งโปรแกรมที่ท่านพัฒนามาอวดกันในวารสารบ้างนะ



รูปที่ 9 แสดงแรงที่มีต่อประจุไฟฟ้าในสนามไฟฟ้า

```
8>
 'Turbo Basic By Arthan Sattayarak

 screen 1:color 0,0
      g=1 :ball$="r2f114d1r4d114f1r2"
   DO until timeoffall = 10
          m1 = int(rnd*5)+1 :q1  = int(rnd*6)+1
          k1 =int(rnd*2) :if k1 = 1 then q1 = -q1
            m2 = int(rnd*5)+1 :q2  = int(rnd*6)+1
            k2 =int(rnd*2) :if k2 = 1 then q2 = -q2
          t = 0 :u1 = 2 :e =1 :u2 = 4
        DO until t > 20
            s1=u1*t + 1/2*(m1*g+q1*e)/m1*t*t +20
            s2=u2*t + 1/2*(m2*g+q2*e)/m2*t*t +20
          draw"bm160,"+str$(int(s1))+"c1"+ball$
          draw"bm167,"+str$(int(s2))+"c1"+ball$:delay .1
            draw"bm160,"+str$(int(s1))+"c0"+ball$
            draw"bm167,"+str$(int(s2))+"c0"+ball$
             t = t+.25
        LOOP
          timeoffall = timeoffall+1
   LOOP
```

รูปที่ 10 โปรแกรมแสดงการเคลื่อนที่ของหยดน้ำมันที่มีประจุไฟฟ้าในสนามไฟฟ้า